**Détection des spams par SMS grâce à l'apprentissage automatique**

# Introduction

Le service de messagerie SMS, plus connu sous le sigle de SMS (pour « Short Message Service ») ou les noms de « texto » ou de « minimessage », permet de transmettre de courts messages textuels. C'est l'un des services de la téléphonie mobile (il a été introduit par la norme GSM). Parmi ces messages (SMS) certains peuvent avoir un but malveillant, on parle alors de SPAM.

Le spam est un courrier indésirable, aussi appelé pourriel, c’est-à-dire un mail ou un SMS non sollicité, envoyé dans un but publicitaire ou malveillant. Il peut simplement consister en une prospection commerciale qui ne respecterait pas les obligations légales de recueil du consentement des destinataires ou carrément tenter d’arnaquer le destinataire, par des astuces frauduleuses pour gagner de l’argent ou en tentant de le faire cliquer sur des liens de phishing. Il s’agit alors de soutirer des données sensibles à l’utilisateur, ou de diffuser des virus.

De nos jours dans le monde numérique que nous vivons, les données à caractères personnelles représentent l’une des informations cruciales dont la protection est une obligation pour chacun. C’est dans ce contexte que des techniques de classification ont été mise en place pour permettre la détection automatique de ces Spams.

Notre travail consistera de ce fait de mettre en place une de ces techniques de détections des Spam par SMS en utilisant des branches de l’intelligence artificielle à savoir le Machine Learning (l’apprentissage automatique) et à l’aide du traitement du langage naturel (NLP).

Notre travail sera étalé suivant plusieurs étapes. Premièrement nous allons faire une revue littéraire des anciennes techniques utilisées afin de pouvoir ressortir les avantages de l’utilisation de l’apprentissage automatique dans ce contexte. En deuxième étape nous allons mettre en évidence le travail effectué sur les données à partir de la collecte jusqu’au prétraitement. Ensuite nous allons passer à la partie de l’apprentissage automatique combiné au traitement du langage naturel dans laquelle nous allons faire notre choix du model à utiliser. En dernière étape nous allons effectuer le déploiement sur une plateforme de test afin de pouvoir mieux utiliser notre application.

# **Revue littérature**

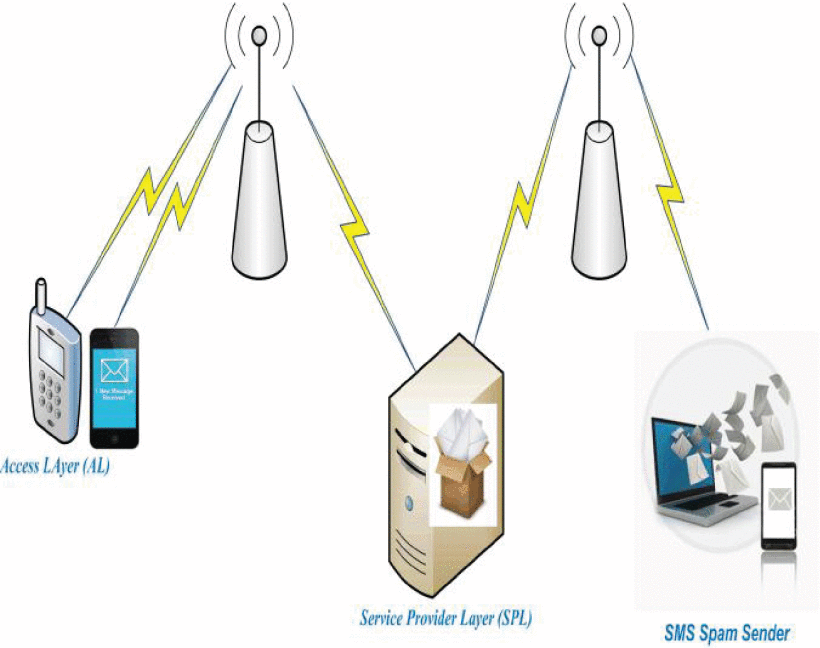
## **Études antérieures sur la détection de spam SMS**

Le spam peut être décrit comme un message électronique non désiré ou non sollicité envoyé en masse à un groupe de destinataires. Les messages sont caractérisés comme étant électroniques, non sollicités, commerciaux, de masse et constituent une menace croissante, principalement en raison des facteurs suivants : 1) la disponibilité de plans de SMS en masse peu coûteux ; 2) la fiabilité (puisque le message atteint l'utilisateur du téléphone mobile) ; 3) la faible probabilité de recevoir des réponses de la part de certains destinataires peu méfiants ; et 4) la possibilité de personnaliser le message. La détection et la prévention du spam par SMS n'est pas une question triviale. Elle a repris un grand nombre de problèmes et de solutions hérités de scénarios relativement anciens de détection et de filtrage du spam par courrier électronique [8]. Les SMS non sollicités sont monnayés courante dans notre vie quotidienne et consomment du temps de communication, de la bande passante et des ressources. Bien que les filtres anti-spam existants offrent un certain niveau de performance, les spams désinforment les destinataires en manipulant des échantillons de données.

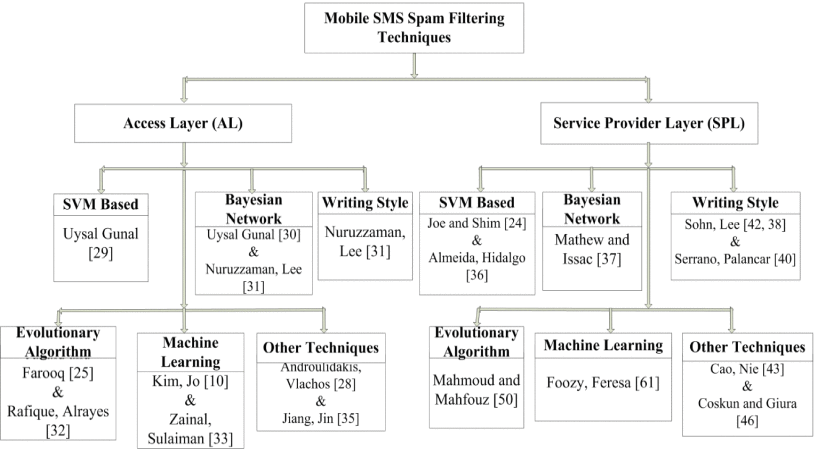
Les études existantes montrent que les techniques de **filtrage** des spams par SMS sont restées à leur stade initial de classification, par exemple la simple similarité des chaînes de caractères ou le blocage explicite des numéros. Les techniques de filtrage traditionnelles telles que le **filtre de classification bayésien**, la régression logistique et l'algorithme de l'arbre de décision pour atténuer les messages de spam SMS prennent encore beaucoup de temps. Des études ont été réalisées sur les différents types de techniques proposées pour le filtrage et l'atténuation du spam SMS mobile. Cependant, il y a encore des progrès à faire en ce qui concerne le filtrage et l'atténuation des spams SMS.

## **Techniques classiques utilisées pour détecter le spam SMS**

Des études récentes sur le spam SMS mobile montrent que plusieurs techniques sont utilisées pour détecter, filtrer ou classer les messages texte de spam. Les solutions sont conçues pour fonctionner soit dans la couche d'accès (AL), soit dans la couche du fournisseur de services (SPL). La couche d'accès est la couche utilisateur, principalement utilisée sous la forme d'un logiciel léger sur la plateforme Android. Un certain nombre de techniques de spamming sont conçues pour fonctionner directement sur le téléphone mobile. D'autres techniques sont conçues pour être déployées au niveau de la SPL. La figure 1 montre l'architecture de base de la ligne de transmission des spams SMS, et la figure 2 présente la taxonomie des recherches sur les spams SMS mobiles.



**Figure 1 : Architecture de la ligne de transmission des spams par SMS.**



**Figure 2 : Taxonomie des techniques de spamming des SMS mobiles.**

### Access Layer (AL)

Cette technique a été mise en application par pas mal d’acteurs de la technologie, nous allons prendre l’exemple de **Junaid** et **Farooq** qui ont appliqué un classificateur d'apprentissage évolutionnaire pour créer un système de détection qui filtre les SMS non sollicités au niveau de la couche d'accès d'un téléphone mobile. Ils étudient un message SMS dans le système hexadécimal et extraient deux caractéristiques de ce format, les bigrammes octet1 et la distribution de fréquence des octets. Ils évaluent la praticabilité d'un certain nombre de classificateurs évolutifs et non évolutifs (fonctionnels sur les ensembles de caractéristiques excédentaires) pour le système de filtrage. Les résultats des expériences suggèrent que le système de classification supervisé (SCS) fonctionnant sur l'ensemble de caractéristiques atteint un taux de détection supérieur à 89 % et un taux de fausse alarme proche de 0 %.

Nous avons aussi Kim et al qui ont mis au point un ratio de fréquence des mots clés et un schéma de simulation de l'outil d'apprentissage automatique WEKA 3.7 pour le système léger et rapide. Le filtre de messages mobiles peut être exécuté de manière autonome dans les téléphones grâce à l'utilisation du rapport de fréquence des mots-clés (FR). Chaque message est décomposé en un ensemble de mots-clés en utilisant la fonction "string to word vector" dans une interface WEKA. Le prétraitement est ensuite effectué sur 5 574 messages SMS.

### Service Providers Layer (SPL)

Joe et Shim [24] ont utilisé un SVM pour filtrer le spam dans le système mobile en appliquant un apprentissage basé sur l'expérience pour identifier les SMS non sollicités. Les termes contenus dans le texte d'un SMS sont extraits en passant par un préprocesseur et un dictionnaire. Si le terme homogénéisé est contenu dans la liste des caractéristiques, le catalogue de mots est fixé à 1 ou 0. Les valeurs vectorielles produites sont utilisées comme données d'apprentissage pour modifier l'hyperplan du SVM. Une fois que chaque vecteur de caractéristiques est marqué 0 ou 1, un processus d'apprentissage est conclu à partir du classificateur SVM latéral. Une fonction de base radiale (RBF) gaussienne est utilisée comme fonction noyau. La valeur constante est fixée à 10, 20, 40, et les valeurs gamma sont fixées à 0,01, 0,05 et 0,1. La technique montre ses performances avec un taux de vecteur de caractéristiques de 150, un taux constant de 20 et un taux de gamma de 0,01. Le taux de détection est considérablement réduit lorsque le dispositif de prétraitement ne peut pas séparer correctement les lignes de mots.

Mahmoud et Mahfouz [50] ont créé un système immunitaire artificiel (SIA) de classification des SMS pour filtrer le spam par SMS. Le système AIS utilise un ensemble de caractéristiques pour servir de filtre anti-spam en entrée. Il catégorise les messages textuels à l'aide d'un ensemble de données formées qui se compose de numéros de téléphone, de mots de spam et de détecteurs. Les résultats expérimentaux sont obtenus en utilisant le système d'exploitation iPhone (iOS). Les résultats de cette expérience montrent que le système proposé est capable de classer les messages en tant que spam ou non-spam avec plus de précision et de vitesse de convergence que l'algorithme bayésien naïf.

## **Avantages de l'utilisation de l'apprentissage automatique dans ce contexte**

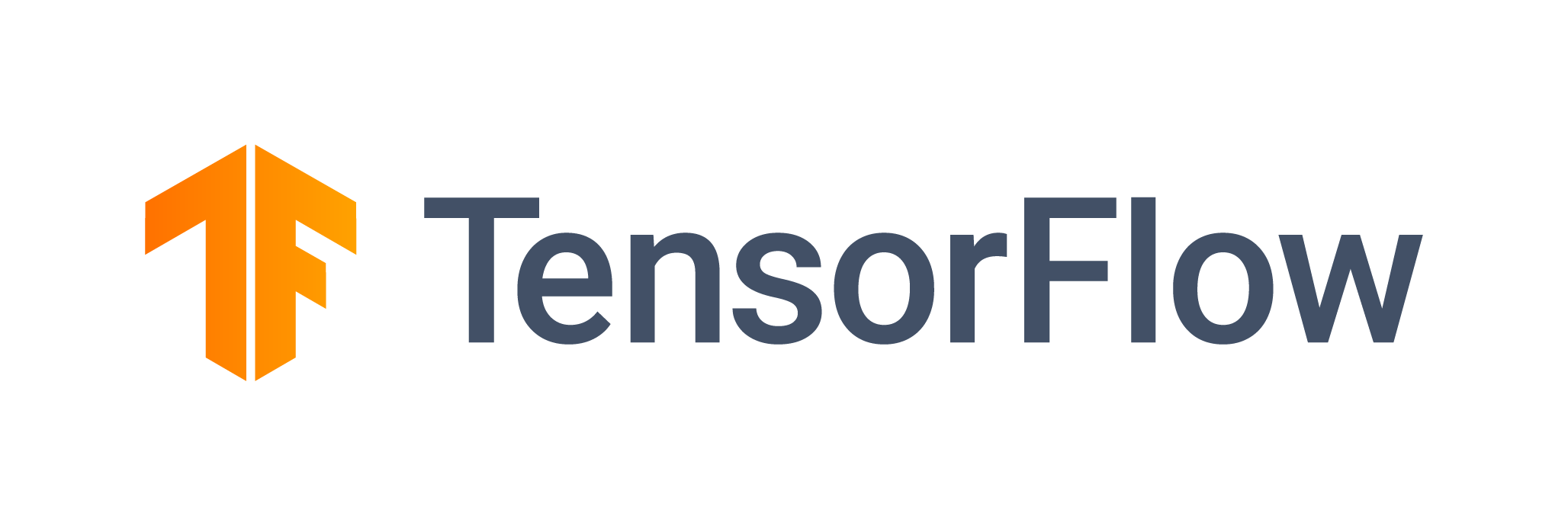
L'apprentissage automatique est une branche de l'intelligence artificielle (IA) et de l'informatique qui porte sur l'utilisation des données et des algorithmes pour imiter la manière dont les êtres humains apprennent, afin d'améliorer progressivement sa précision.

Au cours des deux dernières décennies, les avancées technologiques dans le domaine du stockage et de la puissance de traitement ont permis de créer des produits innovants basés sur l'apprentissage automatique, tels que le moteur de recommandation de Netflix et les voitures autonomes.

L'apprentissage automatique est une composante importante du domaine en pleine expansion qu'est la science des données. Grâce à l'utilisation de méthodes statistiques, des algorithmes sont entraînés à effectuer des classifications ou des prévisions, ce qui permet de découvrir des informations essentielles dans le cadre de projets d'exploration de données. Ces informations permettent ensuite de prendre des décisions dans les applications et les entreprises, et ont idéalement un impact sur les principales mesures de croissance. Avec l'essor du Big Data, la demande de spécialistes des données va continuer d'augmenter. Ces derniers devront identifier les questions économiques les plus pertinentes et les données permettant d'y répondre.

Les algorithmes d'apprentissage automatique sont généralement créés à l'aide d'infrastructures qui accélèrent le développement de solutions, tels que TensorFlow et PyTorch.

**TensorFlow** est un outil open source d'apprentissage automatique développé par Google.



**Figure 3 : TensorFlow**

**PyTorch** est une bibliothèque logicielle Python open source d'apprentissage machine qui s'appuie sur Torch développée par Meta.



**Figure 4 : PyTorch**

L'utilisation de l'apprentissage automatique pour la détection de spam SMS présente plusieurs avantages, notamment une meilleure capacité d'adaptation aux changements dans les modèles de spam, une précision accrue et une réduction du besoin de règles manuelles. Voici quelques-uns des avantages clés :

* **Adaptabilité aux Modèles de Spam Evolutifs :**

Les modèles de spam SMS évoluent constamment. L'apprentissage automatique peut s'adapter dynamiquement aux nouveaux schémas de spam sans nécessiter une mise à jour manuelle constante des règles.

* **Capacité à Gérer des Grandes Quantités de Données :**

L'apprentissage automatique est particulièrement efficace lorsqu'il est confronté à de grandes quantités de données. Les modèles peuvent être entraînés sur d'énormes ensembles de données, ce qui peut améliorer la capacité du modèle à généraliser et à détecter des schémas subtils.

* **Réduction des Faux Positifs :**

En utilisant des modèles d'apprentissage automatique, la détection de spam peut être plus précise, réduisant ainsi le nombre de faux positifs (messages légitimes marqués comme spam). Cela améliore l'expérience de l'utilisateur en évitant la perte de messages importants.

* **Apprentissage à Partir de l'Expérience :**

Les modèles d'apprentissage automatique peuvent apprendre à partir de l'expérience en ajustant leurs poids et leurs paramètres en fonction des erreurs passées. Cela permet une amélioration continue de la précision du modèle au fil du temps.

* **Flexibilité dans le Choix des Caractéristiques :**

Les modèles d'apprentissage automatique offrent la flexibilité de choisir parmi diverses caractéristiques, telles que la fréquence des mots, la longueur du message, la présence de liens, etc., pour améliorer la détection du spam.

* **Réaction Rapide aux Changements :**

L'apprentissage automatique permet une réaction rapide aux changements dans les tendances du spam. Les modèles peuvent être ré-entraînés régulièrement pour rester efficaces dans la détection des schémas de spam émergents.

En vue de ces précédents avantages cités sur l’utilisation de l’apprentissage automatique pour la détection des Spam SMS, nous allons maintenant dans la partie suivante faire une étude détaillée de cette méthode avec un cas pratique à l’appui.

# **Collecte et Analyse des données**

Dans tout processus d’apprentissage automatique les données représentent un facteur primordial. Sans données, aucune opération ne pourrait être effectuée.

Dans cette partie nous allons faire une étude détaillée de nos données allant de la collecte, à l’exploration jusqu’au prétraitement.

## **Collecte des données**

### **Source de données :**

Dans notre cas pratique nous avons utilisé un jeu de données public du nom de **« SMS Spam Collection ».** La collection de SMS Spam est un ensemble de messages SMS étiquetés qui ont été collectés pour la recherche sur les SMS Spam. Elle contient un ensemble de SMS en anglais de 5 572 messages, étiquetés selon deux catégories à savoir "ham" (légitimes) ou "spam".

Les fichiers contiennent un message par ligne. Chaque ligne est composée de deux colonnes : v1 contient l'étiquette (ham ou spam) et v2 contient le texte brut.

Ce corpus a été collecté à partir de sources gratuites ou libres pour la recherche sur Internet :

Une collection de 425 messages de spam SMS a été extraite manuellement du site Web Grumbletext. Il s'agit d'un forum britannique dans lequel les utilisateurs de téléphones portables font des déclarations publiques sur les messages de spam SMS, la plupart du temps sans signaler le message de spam reçu. L'identification du texte des messages de spam dans les réclamations est une tâche très difficile et fastidieuse, qui a nécessité l'analyse minutieuse de centaines de pages web. Le site web de Grumbletext est le suivant [Lien Web].

Un sous-ensemble de 3 375 SMS choisis au hasard parmi les messages de spam du NUS SMS Corpus (NSC), qui est un ensemble de données d'environ 10 000 messages légitimes collectés à des fins de recherche au département d'informatique de l'université nationale de Singapour. Les messages proviennent en grande partie de Singapouriens et surtout d'étudiants de l'université. Ces messages ont été collectés auprès de volontaires qui ont été informés que leurs contributions allaient être rendues publiques. Le corpus SMS de la NUS est disponible à l'adresse suivante : [Lien web].

Une liste de 450 messages SMS ham recueillis dans le cadre de la thèse de doctorat de Caroline Tag est disponible à l'adresse suivante : [Web Link].

Enfin, nous avons intégré le SMS Spam Corpus v.0.1 Big. Il contient 1 002 SMS ham et 322 messages de spam et est public.

Voici un aperçu du jeu de données en question :

|  | **label** | **message** |
| --- | --- | --- |
| 0 | ham | Go until jurong point, crazy.. Available only ... |
| 1 | ham | Ok lar... Joking wif u oni... |
| 2 | spam | Free entry in 2 a wkly comp to win FA Cup fina... |
| 3 | ham | U dun say so early hor... U c already then say... |
| 4 | ham | Nah I don't think he goes to usf, he lives aro... |
| ... | ... | ... |
| 5567 | spam | This is the 2nd time we have tried 2 contact u... |
| 5568 | ham | Will ü b going to esplanade fr home? |
| 5569 | ham | Pity, \* was in mood for that. So...any other s... |
| 5570 | ham | The guy did some bitching but I acted like i'd... |
| 5571 | ham | Rofl. Its true to its name |

**Tableau 1 : Jeu de données SMS Spam**

Une fois le jeu de données choisit, nous aurons ensuite besoin de le charger dans notre IDE afin de pouvoir faire les manipulations nécessaires.

Pour ce faire nous avons fait recours à une bibliothèque python du nom de **Pandas.**

**Pandas** est une bibliothèque écrite pour le langage de programmation Python permettant la manipulation et l'analyse des données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles.

Pandas est un logiciel libre sous licence BSD2. Son nom est dérivé du terme Panel Data (en français "données de panel", un terme d'économétrie pour les jeux de données qui comprennent des observations sur plusieurs périodes de temps pour les mêmes individus). Son nom est également un jeu de mots sur l'expression "Python Data Analysis".



**Figure 5 : Pandas**

## **Exploration des données**

L'exploration de données est une technique assistée par ordinateur utilisée en analyse pour traiter et explorer de grands ensembles de données. Grâce aux outils et méthodes d'exploration de données, les organisations peuvent découvrir des tendances et des relations cachées dans leurs données. L'exploration des données transforme les données brutes en connaissances pratiques. Les entreprises utilisent ces connaissances pour résoudre des problèmes, analyser l'impact futur des décisions commerciales, et augmenter leurs marges bénéficiaires.

L'extraction de règles d'association est le processus qui consiste à trouver des relations entre deux ensembles de données différents, apparemment sans rapport. Les énoncés « si-alors » démontrent la probabilité d'une relation entre deux points de données. Les scientifiques des données mesurent la précision des résultats à l'aide de critères de soutien et de confiance. La prise en charge mesure la fréquence d'apparition des éléments liés dans l'ensemble de données, tandis que la confiance indique le nombre de fois où une déclaration de type "si-alors" est exacte.

### **Classification**

La classification est une technique complexe d'exploration de données qui entraîne l'algorithme ML à trier les données en catégories distinctes. Elle utilise des méthodes statistiques telles que les arbres de décision et les plus proches voisins pour identifier la catégorie. Dans toutes ces méthodes, l'algorithme est préprogrammé avec des classifications de données connues pour deviner le type d'un nouvel élément de données.

Par exemple, les analystes peuvent entraîner le logiciel d'exploration de données en utilisant des images étiquetées de pommes et de mangues. Avec une certaine précision, le logiciel peut alors prédire si une nouvelle image est une pomme, une mangue ou un autre fruit.

### **Clustering**

Le clustering consiste à regrouper plusieurs points de données en fonction de leurs similitudes. Il est différent de la classification car il ne peut pas distinguer les données par catégorie spécifique mais peut trouver des modèles dans leurs similarités. Le résultat de l'exploration de données est un ensemble de clusters où chaque collection est distincte des autres groupes, mais où les objets de chaque cluster sont similaires d'une certaine manière.

Par exemple, l'analyse cluster peut être utile pour les études de marché lorsqu'on travaille avec des données multivariées provenant d'enquêtes. Les spécialistes des études de marché utilisent l'analyse cluster pour diviser les consommateurs en segments de marché et mieux comprendre les relations entre les différents groupes.

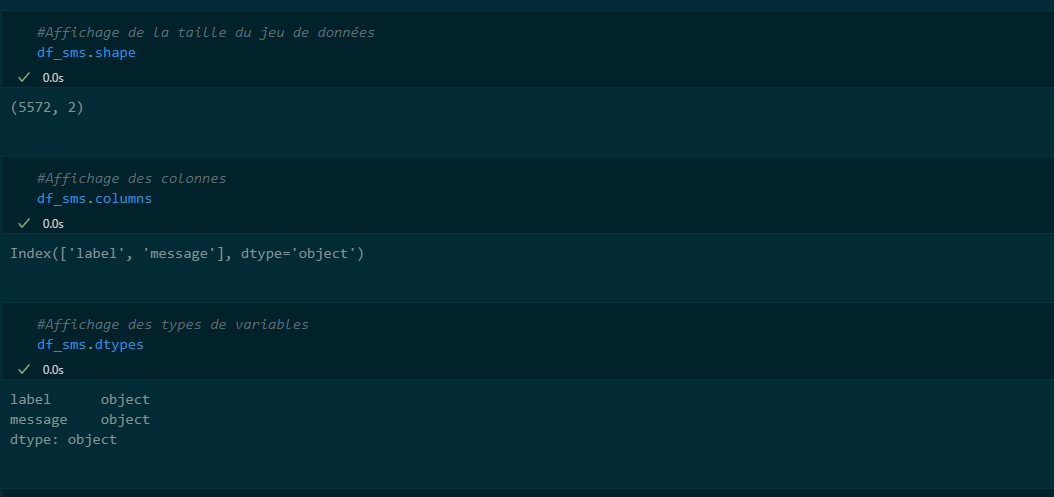
Dans cette partie nous allons procéder à l’exploration de notre jeu de donnée.

### **Description des caractéristiques des données**

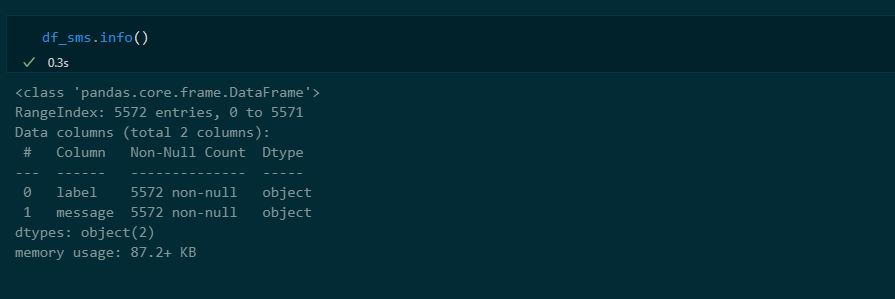
Comme mentionné précédemment le jeu de données est composé de deux colonnes et de **5572** lignes. Les variables sont de types **Object** séparées en deux colonnes : **label** qui représente le type de message (hpam ou spam) et **message** qui contient le texte.

Le jeu de données ne présente ni de valeurs nulles ni de valeurs dupliquées.

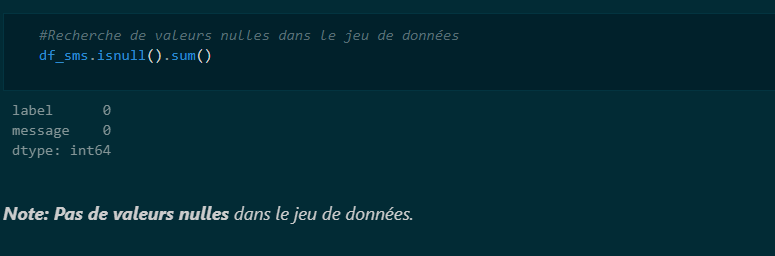
Le processus de recueille de ses informations s’est fait comme suit :



**Capture 1 : Affichage des détails du jeu de données**

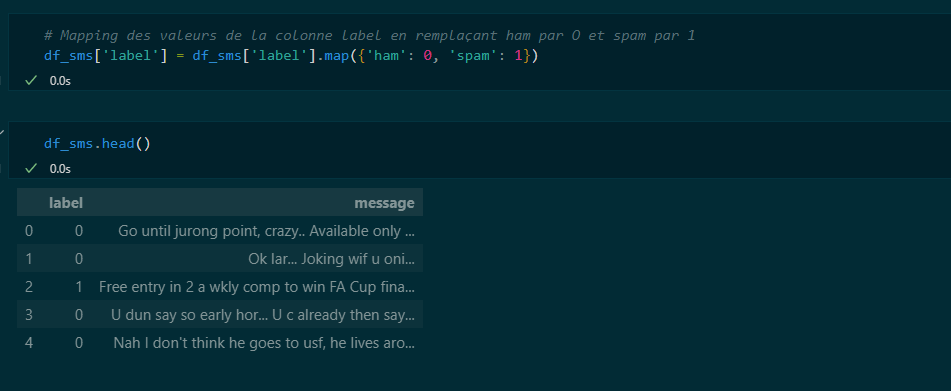


**Capture 1 : Affichage des infos du jeu de données**



**Capture 3 : Vérification de l’existence de valeurs nulles**

Pour une manipulation plus facile de nos données nous avons effectué un **Mapping** au niveau de la colonne **Label** et changer les valeurs ham en 0 et Spam en 1.



**Capture 4 : Mapping de la colonne Label**

### **Statistiques descriptives**

Nous avons ensuite affiché les statistiques descriptives du jeu de données :

|  | **label** | **message** |
| --- | --- | --- |
| count | 5572 | 5572 |
| unique | 2 | 5169 |
| top | ham | Sorry, I'll call later |
| freq | 4825 | 30 |

Tableau 2 : Statistiques descriptives du jeu de données

Interprétation :

**Label** : count (nombre) : Il y a un total de 5572 observations (messages) dans l’ensemble de données.

Unique (unique) : Il y a 2 catégories uniques dans la colonne "label". Cela signifie qu'il y a deux labels possibles. Dans ce cas, les labels sont "ham" (non-spam) et Spam.

top (plus fréquent) : "ham" est la catégorie la plus fréquemment observée dans la colonne "label". Il y a 4825 occurrences de "ham".

freq (fréquence) : La fréquence correspondante à "ham" est de 4825. Cela signifie que "ham" est l'étiquette la plus fréquemment observée dans l'ensemble de données.

**Message** : count (nombre) : Il y a également 5572 observations (messages) dans la colonne "message". Chaque message est associé à un label dans la colonne "label".

Unique (unique) : Il y a 5169 messages uniques dans la colonne "message".

Top (plus fréquent) : "Sorry, I'll call later" est le message le plus fréquemment observé dans l'ensemble de données.

Freq (fréquence) : La fréquence correspondante à ce message est de 30. Cela signifie que ce message particulier apparaît 30 fois dans l'ensemble de données.

### **Graphe de visualisation de la distribution des classes**

Avec le développement des technologies intelligentes qui génèrent des quantités astronomiques de data, la visualisation des données devient indispensable. En effet, pour pouvoir analyse ses Big Data et les utiliser au mieux dans sa stratégie d’entreprise, il est essentiel de pouvoir les lire et répertorier ses informations business dans des tableaux de bord visuels.

La visualisation des données (ou data visualization ou dataviz en anglais) désigne le fait de représenter visuellement ses data pour pouvoir déceler et comprendre des informations, les données brutes étant difficilement interprétables et exploitables. Ce processus se fait par des outils analytiques spécifiques et se matérialise par des tableaux (type Excel), des graphiques, des cartes visuelles ou même des infographies regroupées dans des Dashboard (tableaux de bord).

En classant, segmentant et scénarisant les données visuellement, une entreprise peut déceler des informations auparavant inaccessibles en un simple coup d’œil. La visualisation des données permet donc à toute organisation de piloter son activité plus efficacement en adoptant une stratégie data-driven et agile.

Dans le domaine de la data science plus précisément avec **Python** nous avons utilisées deux principales librairies pour effectuer la visualisation de la distribution de nos données.

La première librairie est **matplotlib.** Matplotlib est une bibliothèque Python open source, initialement développée par le neurobiologiste John Hunter en 2002. L’objectif était de visualiser les signaux électriques du cerveau de personnes épileptiques. Pour y parvenir, il souhaitait répliquer les fonctionnalités de création graphique de MATLAB avec Python.

Suite au décès de John Hunter en 2012, Matplotlib a été améliorée au fil du temps par de nombreux contributeurs de la communauté open source. Elle est utilisée pour créer des graphiques, et des diagrammes de haute qualité. C’est une alternative open source à MATLAB.

Il est par exemple possible de créer des tracés, des histogrammes, des diagrammes à barre et tous types de graphiques à l’aide de quelques lignes de code. Il s’agit d’un outil très complet, permettant de générer des visualisations de données très détaillées.

Cette bibliothèque est particulièrement utile pour les personnes travaillant avec Python ou NumPy. Elle est notamment utilisée sur des serveurs d’application web, des shells et des scripts Python. Avec les APIs de matplotlib, il est aussi possible pour les développeurs d’intégrer des graphiques à des applications d’interface graphique.

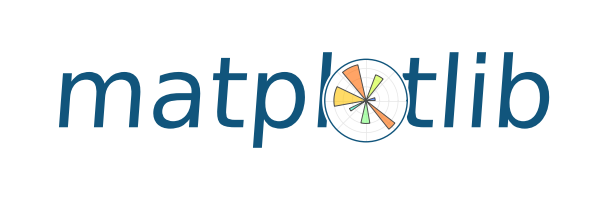


Figure 6 : Matplotlib

La deuxième bibliothèque est **seaborn**. La librairie **Seaborn** vient proposer une alternative à Matplotlib. C’est également une librairie permettant de générer des graphiques, tout comme Matplotlib. Seaborn est exactement pareil vis-à-vis de Matplotlib. La librairie reprend les mêmes principes que Matplotlib, mais en ajoutant quelque chose “par-dessus”, afin d’offrir des qualités de graphiques différents, des nouvelles fonctionnalités, etc. Si vous me permettez une dernière image, on pourrait imaginer que Seaborn est une voiture flambant neuve dont l’ensemble du châssis et le moteur seraient Matplotlib.

Seaborn vient corriger plusieurs défauts de Matplotlib :

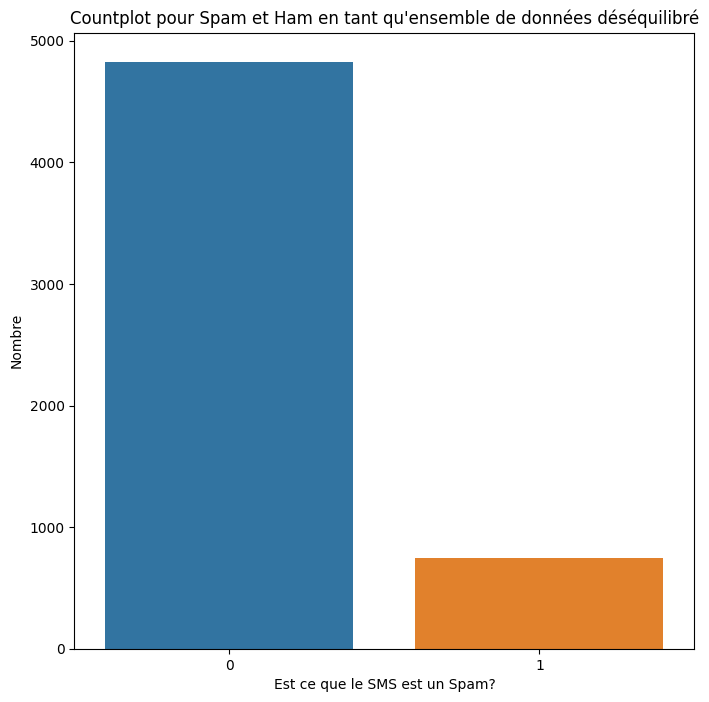
* Il propose de multiples modèles graphiques prédéfinis de bonne qualité esthétique, en modifiant les options graphiques par défaut de Matplotlib ;
* Il ajoute une interaction avec les data frames afin de faciliter grandement la génération de graphiques à partir de ceux-ci ;
* Il propose un catalogue – très – dense de fonctions graphiques pour répondre le plus précisément possible à une problématique donnée.

En contrepartie, Seaborn propose un peu moins d’options de personnalisation en tant que telles.



Figure 7 : Seaborn

L’utilisation de ces deux bibliothèques nous a permis de pouvoir visualiser la distribution de nos données comme le montre la figure ci-après :



**Figure 8 : Graphique de la distribution des données**

En se basant sur ce graphique nous pouvons observer la distribution des données qui est totalement déséquilibré. Nous notons une grande différence entre le nombre de message Ham et les messages SPAM.

Nous allons de ce fait procéder à l’équilibrage du jeu de données afin de bien mener le traitement.

Pour cela nous allons faire recours à certaines techniques comme le **Oversampling** et le **Feature Engineering.**

### **Oversampling**

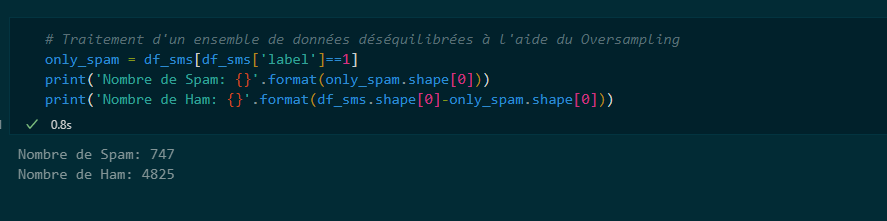
Le suréchantillonnage peut être défini comme le processus consistant à augmenter artificiellement le nombre d'instances dans la classe minoritaire en créant des points de données synthétiques. Ces nouveaux points de données sont générés sur la base des caractéristiques des instances existantes de la classe minoritaire. En introduisant ces instances synthétiques, le suréchantillonnage garantit que les deux classes ont une représentation relativement égale dans l'ensemble de données.

Voyons plus en détail comment fonctionne le suréchantillonnage. Lorsqu'il s'agit d'ensembles de données déséquilibrés, la classe minoritaire a souvent moins d'instances que la classe majoritaire. Ce déséquilibre entre les classes peut conduire à des modèles biaisés qui peinent à prédire avec précision la classe minoritaire. Pour surmonter ce problème, les techniques de suréchantillonnage entrent en jeu.

Une technique de suréchantillonnage couramment utilisée est la technique de suréchantillonnage synthétique des minorités (Synthetic Minority Over-sampling Technique, SMOTE). La technique SMOTE consiste à créer des instances synthétiques le long des segments de ligne reliant les instances de la classe minoritaire. Cette approche garantit que les instances synthétiques se situent dans l'espace des caractéristiques de la classe minoritaire, ce qui les rend représentatives de la distribution sous-jacente.

L'algorithme ADASYN (Adaptive Synthetic Sampling) est une autre technique de suréchantillonnage. ADASYN se concentre sur la génération d'instances synthétiques pour la classe minoritaire qui sont plus difficiles à classer. Ce faisant, ADASYN vise à résoudre le problème du chevauchement des régions entre la classe minoritaire et la classe majoritaire, améliorant ainsi la capacité du modèle à les distinguer.

Afin de procéder à l’équilibrage de la distribution de nos données nous avons opté de faire un Oversampling comme le montre la capture de notre code ci-après :



Explication du code :

Ligne 1 : only\_spam = df\_sms[df\_sms['label']==1]

Cette ligne de code crée un nouveau DataFrame appelé only\_spam qui contient uniquement les lignes où la colonne 'label' a la valeur 1, ce qui indique que le message est spam. En d'autres termes, cela filtre le DataFrame original (df\_sms) pour inclure uniquement les exemples de la classe spam.

Ligne 2 : print('Nombre de Spam: {}'.format(only\_spam.shape[0]))

Cette ligne affiche le nombre de messages spam dans l'ensemble de données en utilisant la fonction shape[0] qui renvoie le nombre de lignes dans le DataFrame only\_spam. La chaîne formatée est utilisée pour inclure cette information dans le message imprimé.

Ligne 3 : print('Nombre de Ham: {}'.format(df\_sms.shape[0]-only\_spam.shape[0]))

Cette ligne affiche le nombre de messages non-spam (ham) dans l'ensemble de données. Elle utilise la propriété shape[0] du DataFrame df\_sms pour obtenir le nombre total de messages, puis soustrait le nombre de messages spam (only\_spam.shape[0]) pour obtenir le nombre de messages ham.

En résumé nous avons effectué une analyse rapide du nombre de message spam et ham.

Ensuite nous avons effectué le suréchantillonnage de la classe minoritaire spam.

count = int((df\_sms.shape[0]-only\_spam.shape[0])/only\_spam.shape[0])

for i in range(0, count-1):

  df\_sms = pd.concat([df\_sms, only\_spam])

df\_sms.shape

Ligne 1 : count = int((df\_sms.shape[0]-only\_spam.shape[0])/only\_spam.shape[0])

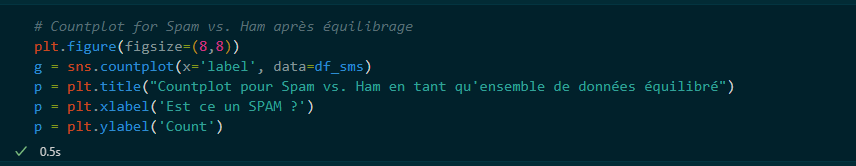
Sur cette ligne nous avons calculé le nombre de fois que les exemples de la classe minoritaure spam doivent être dupliqués pour équilibrer la classe.

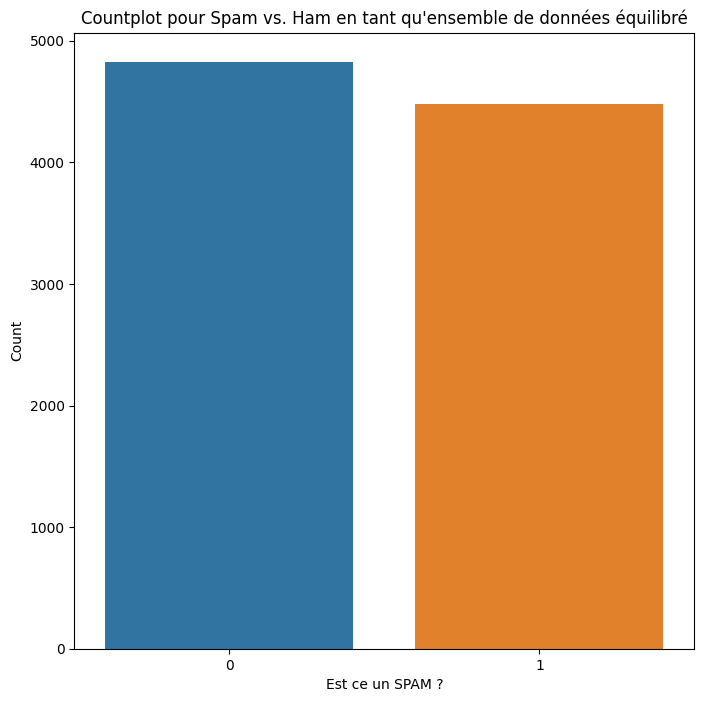
Ligne 2 : for i in range(0, count-1):

  df\_sms = pd.concat([df\_sms, only\_spam])

Nous avons ensuite utilisé une boucle for et à l’aide de la librairie **pandas**, faire une concaténation des jeux de données *df\_sms* et *only\_spam* autant de fois que la valeur calculé dans la ligne précédente.

Afin de pouvoir visualiser les modifications effectuées grâce au suréchantillonnage, affichons le graphique des nouvelles données.





**Figure 8 : Graphique de la distribution des données après suréchantillonnage**

### **Feature Engineering**

Le Feature Engineering est un processus qui consiste à transformer des données brutes en caractéristiques qui représentent plus précisément le problème sous-jacent pour un modèle prédictif

En d'autres termes, il s'agit d'appliquer les connaissances du domaine pour extraire des représentations analytiques des données brutes et les préparer pour l'apprentissage automatique.

Il s'agit de la première étape du développement d'un modèle prédictif d'apprentissage automatique. Elle permet d'accroître la précision du modèle sur des données nouvelles et inédites.

Il est important de se rappeler que les algorithmes d'apprentissage automatique apprennent une solution à un problème à partir d'un échantillon de données. Le Feature Engineering détermine donc la meilleure représentation de l'échantillon de données pour l'apprentissage de la solution au problème.

Ceci est très important car le succès d'un projet d'intelligence artificielle ou d'apprentissage automatique dépend souvent de la représentation des données. Les algorithmes doivent être en mesure de comprendre les données d'entrée.

Le Feature Engineering repose sur un ensemble de procédures et de méthodes bien définies. Les procédures à utiliser varient en fonction des données, et c'est par l'expérience et la pratique que l'on apprend lesquelles utiliser dans un contexte donné.

Voici quelques aspects clés du feature engineering :

**Sélection des Caractéristiques** : Identifiez les caractéristiques les plus importantes pour le problème que vous essayez de résoudre. La sélection judicieuse des caractéristiques peut améliorer l'efficacité du modèle et réduire la complexité.

**Création de Nouvelles Caractéristiques** : Générez de nouvelles caractéristiques à partir des caractéristiques existantes qui peuvent capturer des relations non linéaires ou des informations importantes. Par exemple, créer une caractéristique "ratio" ou "somme" à partir de deux caractéristiques existantes.

**Transformation des Données** : Appliquez des transformations sur les caractéristiques existantes pour les rendre plus adaptées au modèle. Cela peut inclure des opérations comme la normalisation, la standardisation, ou l'application de fonctions mathématiques.

**Gestion des Valeurs Manquantes** : Développez des stratégies pour traiter les valeurs manquantes dans les données. Cela peut impliquer l'imputation (remplacement) des valeurs manquantes par des moyennes, des médianes, ou d'autres stratégies basées sur le contexte.

**Encodage des Catégories** : Convertissez les variables catégorielles en une forme que les modèles d'apprentissage automatique peuvent comprendre. Cela peut inclure l'encodage one-hot, l'encodage ordinal, ou d'autres techniques d'encodage.